## ARGUMENTS/REMARKS

Applicants would like to thank the Examiner for the careful consideration given the present application, and for the personal interview conducted with the Examiner and his supervisor on September 13, 2005. The application has been carefully reviewed in light of the Office Action and interview, and amended as necessary to more clearly and particularly describe and claim the subject matter, which Applicants regard as the invention.

Claims 1–15 remain in this application. The claims have been amended utilizing features found on page 19 of the specification.

Claims 1, 4, 7, 10 and 13 were rejected under 35 U.S.C. §103(a) as being unpatentable over Admitted Prior Art (APA) found on pages 1–2 of the specification. For the following reasons, the rejection is respectfully traversed.

First, a previously stated, Applicant has not admitted that the materials are properly considered prior art under 35 U.S.C. §102.

Second, claim 1, as amended, recites a step performed by an apparatus which "compares program size of the executable objects obtained by the linking processing step with the program size of a executable objects stored in a storing section every time when the linking order is changed" wherein a "minimum program size of the executable object can be determined without any manual intervention being utilized in examining the linking orders of the intermediate objects." Claims 4, 7 and 10 each claim an *apparatus* having component parts with functionality similar to the method of claim 1. Claim 13, as amended, recites a *recording medium* having a *program* for performing similar functions. As discussed at the personal interview, the APA does not teach any such apparatus, its component parts, or such program.

In the Office Action the Examiner cites page 2, lines 5–11 of the application Background section as teaching the comparing step of the claim. However, as discussed in detail at the personal interview, the cited passage does not teach the use of any apparatus for performing the cited functions, and thus does not teach the cited elements of the amended claim 1.

There is no teaching or suggestion in the cited passage for any device or section used for comparing with an executable object that is stored in a storing section. In fact, there is no discussion at all of a storing section. As discussed at the personal interview, the Background section of the application makes clear that the discussion pertains to a *manual* process (see lines 15–18). No apparatus is disclosed. Thus, claim 1 is patentable over the Background section, as are claims 4, 7, 10 and 13 for at least the above reason.

As also discussed at the personal interview, claim 1 also recites that the apparatus has a storing step of "storing the program size of the executable objects and the linking order obtained by the linking processing step in the storing section to update when the program size of the executable objects obtained by the linking processing step is smaller than the program size of the executable objects stored in the storing section at the comparing step." As discussed above, there is no suggestion in the Background material cited by the Examiner of any storing section or storing function. A manual process does not inherently require any such function. Claims 1, 4, 7, 10 and 13 all recited similar limitations. Thus, the cited claims are also patentable over the alleged APA for that reason as well.

In response to prior arguments, the Examiner states in the Office action that "the passage clearly states smaller code size is better and therefore multiple linking orders and executable necessary" (see the second full paragraph on page 8 of the Office Action). Applicant is at a loss as to how the statement "smaller code is better" makes "multiple linking orders and executable necessary." There is no such teaching. Applicant can only surmise that the Examiner is stating that such an implementation would be "obvious" to provide, because no such teaching is found in the Background section of the specification.

The Examiner admits as much on page 3 of the Office Action, stating that the "APA did not explicitly state a storing step for storing the program size of the executable objects and the linking order obtained by the linking processing step in the storing section to update when the program size of the executable objects (sic)."

The Examiner also states that the APA has "demonstrated that it was known at the time of the invention to provide objects to compare as iterations are performed

for optimization," citing page 2, lines 5–11 of the specification. But the cited section says nothing at all about providing objects to "compare as iterations are performed for optimization."

But even if one were to find that the cited portion does provide such a teaching, it is clear that there is no teaching of the *storing step* or *storing section*, as the Examiner has admitted. Instead, the Examiner states that it would be "obvious" to add such a step. But the background discusses only the shortcomings of the manually performed background art, and identifies the problems with that art. It is clear that there is no proper motivation to add any of features that the Examiner states are "obvious," in particular there is no motivation for adding the "storing" step as limited by the claim language.

It is well established that the Examiner cannot look to the application itself for the proper motivation, because such hindsight is impermissible; the facts must be gleaned from the prior art. (MPEP §2142, last paragraph).

The Examiner merely states that it would have been obvious to modify the APA because it would "provide abilities to accomplish the comparison from one moment to the next" citing page 2, lines 5–11. But the Examiner has failed to show how such a desire leads to the *specific* modifications to the APA to get to the invention of claim 1. Such a generalized need hardly leads to specific modifications; many alternative modification may be possible, and thus the Examiner must rely on the teachings of the application to arrive at the specific modifications suggested, which, of course, is hindsight motivation, and that is not allowed. Accordingly, the rejection of claim 1 over the APA is improper, and should be withdrawn.

Furthermore, Applicant's representative notes that claims 4, 7 and 10 are all directed toward an apparatus, and thus cannot read on a manual process. A manual process will not have any storing sections, comparing sections, etc. And claim 13 recites a recording medium that is also not required. Thus, those claims clearly cannot read on the manual process identified in the Background by the Examiner.

In particular, claim 4 is directed toward an "intermediate object linking *unit* for linking a plurality of intermediate objects to form an executable object." The Examiner attempts to point to the APA for teaching such a unit. However, as

discussed above, the Background section makes clear that it is discussing a *manual* method (see page 2, lines 15–18). Thus, there can be no (and there is no) teaching of the "intermediate object linking order forming *section* which decides linking orders of the plurality of intermediate objects" or a "linker starting *section* which executes linking processes of the plurality of intermediate objects based on the linking orders decided by the intermediate object linking order forming section to form an executable object." Thus, claim 4 is patentable over the cited APA.

Furthermore, claim 4 recites a storing *section* similar to claim 1, and thus is patentable over the alleged APA for the corresponding reasons discussed for claim 1.

Furthermore, claim 4 recites that "the storing section stores the program size of the executable objects and the linking order formed by the linker starting section when the program size of the executable objects formed by the linker starting section is smaller than the program size of the executable objects stored in the storing section at the comparison by the comparing section." There is no suggestion of any storing section in the alleged APA, and there is accordingly no suggestion of storing according to the cited criteria. Thus, claim 4 is patentable over the alleged APA for this reason as well.

Claim 7 is similar in that it recites a "linker unit for linking a plurality of intermediate objects to form an executable object" with similar structure as defined in claim 4.

The Examiner is attempting to take a discussion of a *manually* operated method disclosed in the Background, and state that it would be obvious to make the *units* and/or *sections* of claims 4 and 7. This is clearly not appropriate. The Examiner provides no real explanation of how it would be obvious to create the units of claims 4 and 7 from the limited discussion of the manual method disclosed in the Background section. The teachings do not support any such attempt.

Furthermore, the Examiner attempts to provide generalized motivation for modifying the manual method of the Background, but such motivation clearly relies on the disclosure itself, and thus is improper hindsight motivation. Accordingly, claims 4 and 7 are patentable over the so-called APA.

Claim 10 recites a "compiler driving *unit* for translating a source program by starting a compiler, an assembler, a linker, etc. to form an executable object" having one or more limitations similar to those discussed above, and thus are patentable over the alleged APA for at least the same reasons.

Furthermore, claim 13 recites a "recording medium for recording a program for linking a plurality of intermediate objects to form an executable object, wherein the program for causing a computer to execute a method." The Background suggests no such recording medium or program, and a manual method cannot teach such a program. Thus, claim 13 is patentable over the supposed APA as well.

Claims 1, 4, 7, 10 and 13 were rejected under 35 U.S.C. §103(a) as being unpatentable over Smaalders *et al.* (U.S. 5,790,865). For the following reasons, the rejection is respectfully traversed.

As discussed above, claim 1 recites a

> comparing step which compares program size of the executable objects obtained by the linking processing step with the program size of a executable objects stored in a storing section every time when the linking order is changed;

Claim 13 recites similar limitations at lines 13–17. Furthermore, the claims, as amended, recite that "a minimum program size of the executable object can be determined without any manual intervention being utilized in examining the linking orders of the intermediate objects." The reference does not teach finding a minimum program size.

As previously argued, the Examiner admits that the cited reference does not teach such a comparing step. Instead, the Examiner states that it was known to "work toward smaller code" and that it would then be obvious to add the comparing step of the invention to Smaalders. Thus, the Examiner appears to be arguing that, because the result of the method of the invention may be known or desired, that the method would be obvious. Applicant strongly disputes that this is a proper rejection. Just because smaller code size might be desirable does not lead, by itself, to the invention. The Examiner is again reminded that claims 1 and 13 recite a process/method. As such, the burden is on the Examiner to show that the method is

either known in the art, or legally obvious. It is not relevant that the method/process may share a common goal or benefit with another method/process or device. The Examiner is required to show that the process itself is known or obvious. This the Examiner has failed to do.

Furthermore, the reference does not teach that a minimum program size is the goal of the disclosed apparatus. Instead, the reference makes clear that its goal is to reorder code section "based on when the code sections were accessed during the execution period" (col. 2, lines 57–60). One skilled in the art would understand that this goal is not the same as finding a "minimum code size," and in fact, could contradict such a goal. Furthermore, the reference teaches monitoring how often a code section is accessed, and then reordering the sections based on that information (col. 3, lines 1–5). This is a very different method of reorganizing code compared to the method of the claims, and in fact, teaches away from the claimed method.

As previously argued, there is nothing in the Smaalders reference that suggests the comparing step of the claim, because code size is not determinative for Smaalders. Instead, as discussed above, Smaalders teaches a method whereby software code sections are monitored for frequency of access/execution, and the code rearranged based on that access (see col. 2, line 61 to col. 3, line 11). There is nothing in the reference that suggests the comparing step of claims 1 and 13. Accordingly, claims 1 and 13 are patentable over the reference.

In addition, claim 1 recites the step of:

> a storing step for storing the program size of the executable objects and the linking order obtained by the linking processing step in the storing section to update when the program size of the executable objects obtained by the linking processing step is smaller, than the program size of the executable objects stored in the storing section at the comparing step;

Yet again, there is no teaching or suggestion in the reference for determining or storing the program *size*. Therefore, claims 1 and 13 are patentable over the reference for this reason as well.

Claims 4, 7 and 10 recite structure(s) that perform the method steps cited for claims 1 and 13, above. The reference fails to teach either the method steps, or

structure for performing the method steps. Accordingly, claims 4, 7 and 10 are patentable over the reference.

In response to the above argument, the Examiner states in the Office action that "Smaalders [provides] for *indirectly* comparing code size in so much as comparing speed, as faster code is smaller and smaller code is faster" (emphasis added). Even if true (although one skilled in the art would know that this is not generally true), it is not relevant. The step of the claim clearly states comparing size, not speed. If the prior art teaches comparing speed, it cannot be cited as teaching comparing size. They are different method steps. The fact that they might lead to a similar or identical result does not matter. It is well-known that two different methods that lead to the same result are each separately patentable if not disclosed in the prior art. The Examiner must show the *claimed* step, not some *other* step, as the Examiner has actually done.

Furthermore, one skilled in the art would know that faster code is not necessarily smaller, and smaller code is not necessarily faster. The Examiner's statement is so clearly not generally true that Applicant sees no need to try to disprove it. Instead, if the Examiner wants to rely on such a (clearly false) "fact," he should provide references to back it up.

Even further, the Examiner misreads the reference. Smaalders does not teach any desire for smaller code size. Instead, Smaalders is concerned with the order in which routines are executed, because it desires that the most often accessed routines end up stored in RAM memory, rather than on the hard drive as virtual memory (see col. 1, lines 15–35). Smaalders desires to reduce the amount of RAM required, without a large impact on execution speed, and thus desires to reorder executables in order to keep the most often accessed routines in RAM (see col. 9, lines 12–20). There is no discussion at all about reducing executable *size*. Thus, the reference is not even relevant to the invention. Accordingly, claims 1, 4, 7, 10 and 13 are patentable over the reference.

Finally, yet again, the Examiner has failed to provide the proper motivation for modifying the reference as suggested in the Office Action. In fact, because the Examiner argues that the reference already teaches a method of reducing code size

(although it does not), one taking the Examiner's position would have no motivation for pursuing the totally different such method taught by the invention. Applying the method of the invention would change the principle of operation of the Smaalders device, and that is not permissible, because the proposed modification cannot render the prior art unsatisfactory for its intended purpose, or change the principle of operation of a reference (MPEP §2143.01). If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125. Thus, the rejections for obviousness are not proper and should be withdrawn.

Claims 2–3, 5–6, 8–9, 11–12 and 14–15 were rejected under 35 U.S.C. §103(a) as being unpatentable over the APA cited above, in view of Goldberg ("Genetic Algorithms"). For the following reasons, the rejection is respectfully traversed.

First, Goldberg does not overcome the shortcomings discussed above for the APA, and thus claims 2–3, 5–6, 8–9, 11–12 and 14–15 are patentable over the APA for at least the same reasons as their parent claims discussed above.

Second, in the Examiner's provided "Notice of References Cited," the Examiner lists the document "David E. Goldberg, Genetic algorithms," as being last modified on September 30, 2003. The current application was filed on July 26, 2001. The Examiner has not shown that the cited teaching was known prior to the cited "last modified" date. Accordingly, the reference is not properly cited as prior art under any section of 35 U.S.C. §102, and thus the rejection is clearly improper.

Furthermore, the Examiner cites Goldberg as teaching that it was "known to use permutation and/or genetic algorithms for processing a volume of information." However, the Examiner fails to provide the proper motivation for using permutation and/or genetic algorithms in the manner recited in the claims. The fact that a generalized analysis method may be known is not enough to make its use in a specific implementation obvious. There could be a myriad of other analysis methods that might be instead chosen. The Examiner must provide *particular* motivation for applying the *specific* known technique to the particular use disclosed by the

invention. Instead, the Examiner merely states that it would be obvious to provide known sorting processing algorithms because it would be easily implemented. This is not legally sufficient motivation. Thus, the rejection is not proper.

In consideration of the foregoing analysis, it is respectfully submitted that the present application is in a condition for allowance and notice to that effect is hereby requested. If it is determined that the application is not in a condition for allowance, the Examiner is invited to initiate a telephone interview with the undersigned attorney to expedite prosecution of the present application.

If there are any additional fees resulting from this communication, please charge same to our Deposit Account No. 16-0820, our Order No. 33587.

Respectfully submitted,
PEARNE & GORDON, LLP

By: _____
Robert F. Bodi – Reg. No. 48,540

1801 East 9th Street
Suite 1200
Cleveland, Ohio 44114-3108
(216) 579-1700

Date: October 18, 2005